Reuse Working Group
Breakout #4 – Reuse Readiness Levels

8th Earth Science Data Systems Working Group Meeting

Wilmington, Delaware

October 20–22, 2009

Overview of RRLs

Introduction to RRLs

- Having a measure of the reusability of an asset:
  - Provides potential reusers with additional information about the reuse maturity of the asset:
    - Lets them know what they're getting
    - Gives them a basic feel for what modifications may be needed
  - Helps potential reusers make better informed choices about:
    - What to reuse
    - What best meets their needs
- This measure can be used as a piece of metadata for assets placed in the proposed RES (or anywhere else).
- In an iterative process, volunteers from the WG:
  - Wrote an initial set of levels for each topic,
  - Drafted summaries of each RRL, looking across all topic areas at each level,
  - Created a set of summary RRLs with descriptions by combining information from all topics at the same level, and
  - Made suggested revisions to RRLs and topic area levels based on feedback received from the community.
- The WG is reviewing/revising use cases that have been developed.
- The WG plans to use the RRLs to assess some reusable assets and use the results to revise the levels as needed to ensure consistent assessments can be made.

RRL Topic Areas and Levels

- Topic areas included:
  - Documentation
  - Extensibility
  - Licensing
  - Modularity
  - Packaging
  - Portability
  - Standards compliance
  - Support
  - Verification/Testing
- A scale of 1–9 was used to match the Technology Readiness Level (TRL) scale.
- Topic levels were combined into a single RRL scale.
- A sample RRL calculator was also developed (currently only on the prototype RES) as a means to help providers and consumers rate the reusability of software assets by calculating weighted averages of topic area levels.

Example from Testing/Verification

RRL 4 – Software application tested and validated in laboratory environment

Following successful testing of inputs and outputs, the testing would include integrating an application to establish that the “pieces” will work together to achieve concept-enabling levels. This validation must be devised to support the concept that was formulated earlier and should also be consistent with the requirements of potential system applications. The validation is relatively “low-fidelity” compared to the eventual system: it could be composed of ad hoc discrete components in a laboratory; for example, an application tested with simulated inputs.

NATIONAL AERONAUTICS
AND SPACE ADMINISTRATION

Current RRL Topic Area Levels

	Documentation	Extensibility	Intellectual Property Issues	Modularity	Packaging	Portability	Standards Compliance	Support	Verification and Testing
Level 1	Little or no internal or external documentation available	No ability to extend or modify program behavior	Product developers have been identified, but no rights have been determined.	Not designed with modularity	Software or executable available only, no packaging	The software is not portable	No standards compliance	No support available	No testing performed
Level 2	Partially to fully commented source code available	Very difficult to extend the software system, even for application contexts similar to the original application domain	Developers are discussing rights that comply with their organizational policies.			Some parts of the software may be portable	No standards compliance beyond best practices	Minimal support available	Software application formulated and unit testing performed
Level 3	Basic external documentation for sophisticated users available	Extending the software is difficult, even for application contexts similar to the original application domain	Rights agreements have been proposed to developers.	Modularity at major system or subsystem level only	Detailed installation instructions available	The software is only portable with significant costs	Some compliance with local standards and best practices	Some support available	Testing includes testing for error conditions and proof of handling of unknown input
Level 4	Reference manual available	Some extensibility is possible through configuration changes and/or moderate software modification	Developers have negotiated on rights agreements.			The software may be portable at a reasonable cost	Standards compliance, but incomplete and untested	Moderate systematic support is available	Software application demonstrated in a laboratory context
Level 5	User manual available	Consideration for future extensibility designed into the system for a moderate range of application contexts; extensibility approach defined and at least partially documented	Agreement on ownership, limited reuse rights, and recommended citation.	Partial segregation of generic and specific functionality	Software is easily configurable for different contexts	The software is moderately portable	Standards compliance with some testing	Support provided by an informal user community	Software application tested and validated in a laboratory context
Level 6	Tutorials available	Designed to allow extensibility across a moderate to broad range of application contexts, provides many points of extensibility, and a thorough and detailed extensibility plan exists	Developer list, recommended citation, and rights statements have been drafted.			The software is portable	Verified standards compliance with proprietary standards	Formal support available	Software application demonstrated in a relevant context
Level 7	Interface guide available	Demonstrated to be extensible by an external development team in a similar context	Developer list and limited rights statement included in product prototype.	Clear delineations of specific and reusable components	OS detect and auto-build for supported platforms	The software is highly portable	Verified standards compliance with open standards	Organized/defined support by developer available	Software application tested and validated in a relevant context
Level 8	Extension guide and/or design/developers guide available	Demonstrated extensibility on an external program, clear approach for modifying and extending features across a broad range of application domains	Recommended citation and intellectual property rights statement included in product.				Verified standards compliance with recognized standards	Support available by the organization that developed the asset	Software application "qualified" through test and demonstration (meets requirements) and successfully delivered
Level 9	Documentation on design, customization, testing, use, and reuse is available	Demonstrated extensibility in multiple scenarios, provides specific documentation and features to build extensions which are used across a range of domains by multiple user groups	Statements describing unrestricted rights, recommended citation, and developers embedded into product.	All functions and data encapsulated into objects or accessible through web service interfaces	Installation user interface provided	The software is completely portable	Independently verified standards compliance with recognized standards	Large user community with well-defined support available	Actual software application tested and validated through successful use of application output

Current Draft RRLs

Level	Summary	Description
RRL 1	Limited reusability; the software is not recommended for reuse.	Little is provided beyond limited source code or pre-compiled, executable binaries. There is no support, contact information for developers or rights for reuse specified, the software is not extensible, and there is inadequate or no documentation.
RRL 2	Initial reusability; software reuse is not practical.	Some source code, documentation, and contact information are provided, but these are still very limited. Initial testing has been done, but reuse rights are still unclear. Reuse would be challenging and cost-prohibitive.
RRL 3	Basic reusability; the software might be reusable by skilled users at substantial effort, cost, and risk.	Software has some modularity and standards compliance, some support is provided by developers, and detailed installation instructions are available, but rights are unspecified. An expert may be able to reuse the software, but general users would not.
RRL 4	Reuse is possible; the software might be reused by most users with some effort, cost, and risk.	Software and documentation are complete and understandable. Software has been demonstrated in a lab on one or more specific platforms, infrequent patches are available, and intellectual property issues would need to be negotiated. Reuse is possible, but may be difficult.
RRL 5	Reuse is practical; the software could be reused by most users with reasonable cost and risk.	Software is moderately portable, modular, extendable, and configurable, has low-fidelity standards compliance, a user manual, and has been tested in a lab. A user community exists, but may be a small community of experts. Developers may be contacted to request limited rights for reuse.
RRL 6	Software is reusable; the software can be reused by most users although there may be some cost and risk.	Software has been designed for extensibility, modularity, and portability, but software and documentation may still have limited applicability. Tutorials are available, and the software has been demonstrated in a relevant context. Developers may be contacted to obtain formal statements on restricted rights or to negotiate additional rights.
RRL 7	Software is highly reusable; the software can be reused by most users with minimum cost and risk.	Software is highly portable and modular, has high-fidelity standards compliance, provides auto-build installation, and has been tested in a relevant context. Support is developer-organized, and an interface guide is available. Software and documentation are applicable for most systems. Brief statements are available describing limited rights for reuse and developers may be contacted to negotiate additional rights.
RRL 8	Demonstrated local reusability; the software has been reused by multiple users.	Software has been shown to be extensible, and has been qualified through test and demonstration. An extension guide and organization-provided support are available. Brief statements are available describing unrestricted rights for reuse and developers may be contacted to obtain formal rights statements.
RRL 9	Demonstrated extensive reusability; the software is being reused by many classes of users over a wide range of systems.	Software is fully portable and modular, with all appropriate documentation and standards compliance, encapsulated packaging, a GUI installer, and a large support community that provides patches. Software has been tested and validated through successful use of application output. Multiple statements describing unrestricted rights for reuse and the recommended citation are embedded into the product..

Summary of RRL Use Cases

Current Draft RRL Use Cases

Software Contribution

1. Planning Software Development
2. Identifying Software Development Components to be Created
3. Identifying Standards for Creating Software Development Components
4. Creating Deliverables of Software Development Teams
5. Evaluating Individual Software Components for Integration
6. Integrating Software Components

Software Adoption

1. Planning Software Adoption
2. Identifying Criteria to Assess Software Candidates for Adoption
3. Identifying Potential Software Candidates for Adoption
4. Assessing the Cost of Software Components for Adoption
5. Evaluating the Quality of Software Components for Adoption
6. Determining Progress on Integrating Adopted Components

Software Sponsorship

1. Planning Software Sponsorship
2. Determining the Cost of Software Projects to be Sponsored
3. Establishing Criteria to Assess Software Proposals for Sponsorship
4. Identifying Potential Software Proposals for Sponsorship
5. Evaluating the Quality of Software Proposals for Sponsorship
6. Evaluating Progress of Sponsored Software Projects

Points for Discussion

- Feedback on the categories of the use cases?
- Feedback on the use cases mentioned?

RRL Calculator Tool

Background

- Earlier discussions on RRLs indicated using a (weighted) average of topic area levels could be a useful way of determining an overall RRL for a reusable asset.
- The WG thought an RRL calculator would be a useful tool, so this was developed.
- The RRL calculator web page:
  - Is written in PHP
  - Provides descriptions of the topic area levels and overall RRLs to assist users in making assessments
- Input: topic area level scores (1–9) and optionally a weighting factor (0–100, 0=do not use) for each topic area
- Output: a possibly weighted average of the topic area level scores, which can be used as the overall RRL

Screenshots

Topic *(descriptions below)*	**Score**(1-9)		**Weight**(0-100)
Portability: *The software is highly portable*	7	X	100
Extensibility: *Designed to allow extensibility across a moderate to broad range of application contexts, provides many points of extensibility, and a thorough and detailed extensibility plan exists*	6	X	100
Documentation: *User manual available*	5	X	100
Support: *Moderate systematic support is available*	4	X	100
Packaging: *OS-detect and auto-build for supported platforms available*	7	X	100
IP Issues: *Developer list, recommended citation, and rights statements have been drafted*	6	X	100
Standards Compliance: *Standards compliance with some testing*	5	X	100
Verification and Testing: *Software application tested and validated in a relevant context*	7	X	100
Modularity: *Partial segregation of generic and specific functionality*	5	X	100

Overall RRL: 5.8

RRL 6: Software is reusable; the software can be reused by most users although there may be some cost and risk.
Software has been designed for extensibility, modularity, and portability, but software and documentation may still have limited applicability.
Tutorials are available, and the software has been demonstrated in a relevant context. Developers may be contacted to obtain formal statements on restricted rights or to negotiate additional rights.

As scores and weights are changed, the short description following the topic title changes as does the overall RRL calculated by (weighted) average.

RRL Levels for Packaging

RRL 1: *Software or executable availabe only, no packaging*
Inadequate or no documentation and no auto-build/install facility is available.

RRL 2: none

RRL 3: *Detailed installation instructions available*
System includes auto-build feature, but is built for a particular operating system

RRL 4: none

RRL 5: *Software is easily configurable for different contexts*
For example, locations of resources (files, directories, URLs) are configurable. All configuration-specific information is centralized

RRL 6: none

RRL 7: *OS-detect and auto-build for supported platforms available*
Operating system detection configuration files are available. Packaging includes auto-build for supported OS platforms and suite of regression tests for platform-specific testing.

RRL 8: none

RRL 9: *Installation user interface provided*
A user interface guides the installer through all steps needed to build, configure, and install the software.

Longer descriptions of topic area levels are displayed by clicking a topic area title

Points for Discussion

- How can we improve this RRL calculator tool?
- Is it a useful/valuable tool, something we should continue to work on?
- Can (or should) it be copied/ported to other web sites for use on other systems?

Process for Assessing Assets with RRLs

Idea Behind Assessments

- RRLs and topic area levels have been revised over a long period of time based on community feedback.
- RRLs are intended to be used as a measure, and assessments are meant to ensure precision of measurement.
  - This includes consistency of topic areas across a given level and consistency of assessments by different people.
- Plan to use some other work to inform possible revisions to topic area levels
  - Example: packaging/distribution work for packaging topic area
- Also plan to perform assessments of some reusable assets to check if RRLs need revisions to ensure consistent assessments.
- Initial idea is for a four-step process:
  - Selecting reusable assets to assess
  - Performing the assessments
  - Analyzing the results of the assessments
  - Revising the RRLs as necessary

Discussion

- Selecting reusable assets to assess
  - What types of assets should we assess?
  - How do we select appropriate assets for refining the RRLs?
- Performing the assessments
  - How do we ensure that not having a reuse purpose in mind as an end result will not affect the assessments?
  - Should we provide any guidance on how to do assessments?
- Analyzing the results of the assessments
  - How consistent were the results of different reviewers?
  - How can the RRLs be updated to achieve better consistency?
- Revising the RRLs as necessary
  - How can we use cost/effort, as a fraction or percentage of the total work, to help distinguish or move between RRLs?

Backup Slides